

**UNIVERSIDAD COMPLUTENSE DE MADRID**

**Facultad de Ciencias Físicas**



**Espectros de alta resolución de  
estrellas frías: base de datos para  
clasificación espectral, parámetros  
estelares y abundancias.**

**Jesús Ramos Medina**

**Trabajo Académicamente Dirigido**

Director: David Montes

Septiembre 2012



# ÍNDICE

1. Introducción.	Pág.4
1.1 Espectros <i>échelle</i> de alta resolución.	Pág.4
1.2 Procesado de espectros.	Pág.5
1.3 Programa IRAF.	Pág.6
1.4 Base de datos Saada.	Pág.6
2. Tratamiento de espectros.	Pág.7
2.1 Supresión de rayos cósmicos.	Pág.7
2.2 Normalización al continuo.	Pág.10
2.3 Corte de los órdenes de los espectros.	Pág.12
2.4 Pegado de los órdenes de los espectros.	Pág.14
2.5 Corrección de la velocidad radial.	Pág.15
3. Base de datos.	Pág.21
4. Bibliografía.	Pág.24
5. Apéndice.	Pág.26
5.1 Lista de estrellas.	Pág.26
5.2 Scopy_varios_N.	Pág.27
5.3 Scombine_varios_N.	Pág.40
5.4 Epar continuum.	Pág.40
5.5 Epar dopcor.	Pág.42

## 1. INTRODUCCIÓN.

### 1.1 Espectros *échelle* de alta resolución.

Para obtener el espectro de un objeto astronómico debemos hacer pasar la luz recogida por el telescopio a través de un espectrógrafo.

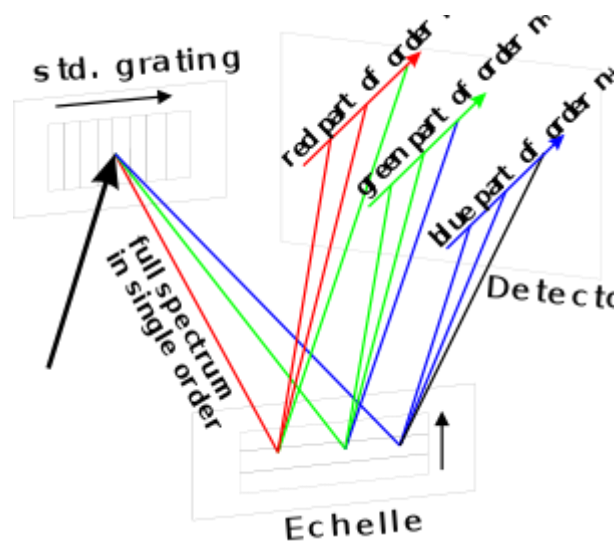
Los espectrógrafos son instrumentos equipados con una red de difracción o un prisma que se encargan de dispersar la luz y producir el espectro deseado. Este espectro queda recogido como imagen en un CCD y puede ser extraído posteriormente con los programas apropiados.

La región del espectro que podemos obtener al medir está limitada por el tamaño del detector, en nuestro caso, el CCD. Además una rápida inspección de las imágenes de espectros tomadas con los CCDs nos muestra una gran zona del CCD sin utilizar.

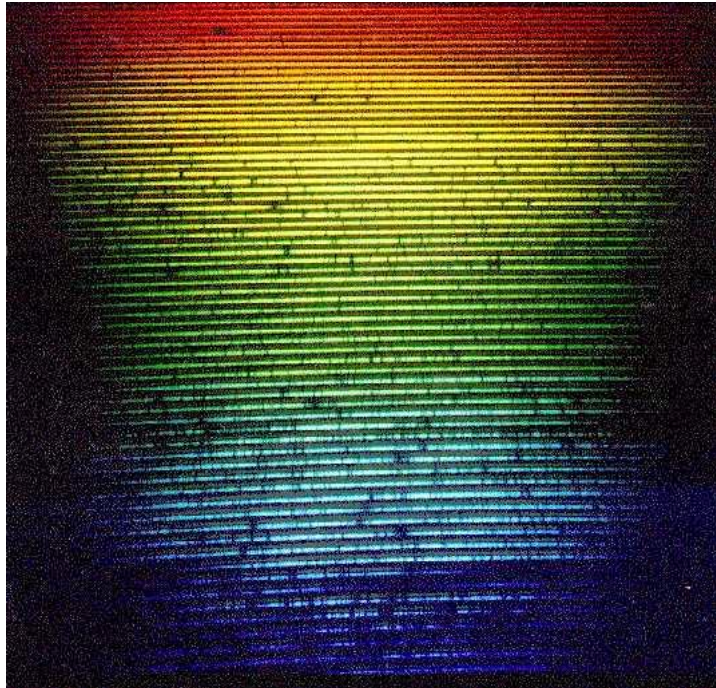
Para optimizar el tamaño del CCD e incrementar su aprovechamiento se introducen los espectrógrafos *échelle*.

Un espectrógrafo *échelle* está formado por una red de difracción que posee un paso de red pequeño lo que nos permite obtener una mayor resolución. Esto produce una gran dispersión en el espectro, pero sólo para un pequeño rango de longitudes de onda dentro de cada orden. A pesar de que el rango dispersado es pequeño, se produce un fenómeno de solapamiento a altos órdenes. Para evitar esto se introduce un segundo elemento de dispersión cruzado que permite separar los órdenes y distribuirlos por todo el área del CCD.

En la siguiente imagen se puede entender de forma más clara el camino recorrido por la luz desde que entra al espectrógrafo hasta que llega al detector (CCD) tras atravesar las dos redes de difracción perpendiculares entre sí.



La imagen del espectro recogida en el CCD sería similar a la siguiente.



## 1.2 Procesado de espectros.

Como ya hemos comentado los espectrógrafos de tipo échelle poseen una configuración que permite dividir el espectro de un objeto en órdenes que ocupan todo el CCD.

La reducción de los datos para extraer el espectro se realiza en tres pasos: en primer lugar, la imagen debe ser corregida de la señal pedestal (BIAS), la corriente de oscuridad (DARK), la luz difusa del espectro y los defectos cosméticos del CCD, así como del patrón de interferencias (fringing).

En segundo lugar es necesario extraer el espectro de cada uno de los órdenes producido por la red échelle y, posteriormente, realizar una calibración en longitud de onda para pasar la escala de píxeles a longitudes de onda.

Por último, el espectro debe ser normalizado al continuo en cada orden para eliminar la forma que produce el espectrógrafo sobre la señal. Esta normalización se realiza ajustando un polinomio de tipo *spline cúbico* al continuo observado en cada uno de los órdenes por separado. Los puntos utilizados para el ajuste se han elegido con el fin de reducir al máximo las zonas de solapamiento de líneas. Dividiendo el espectro del objeto orden a orden por su función ajustada se obtiene un espectro plano con el flujo

continuo situado en la unidad. De esta forma es posible comparar la intensidad de las distintas líneas, así como medir anchuras equivalentes.

Este último paso de normalización al continuo ha sido el que yo he realizado en todos los espectros con los que he tratado. En la segunda parte de este trabajo se explicará de forma más detallada todos los pasos realizados.

Los espectros utilizados en este trabajo fueron tomados en diversas campañas desde el año 1999 hasta el 2009, utilizando el espectrógrafo FOCES (Fibre Optics Cassegrain Echelle Spectrograph) situado en el observatorio Hispano-Alemán de Calar Alto (CAHA) en Almería. El rango de longitudes de onda obtenido con este instrumento va desde los 3910 Å hasta los 9075 Å con una resolución espectral de 0.09 Å a 0.26 Å.

Las estrellas estudiadas son de los tipos espectrales F, G, K y M. En el apéndice final adjuntamos una tabla con todas las estrellas con las que hemos trabajado a lo largo del trabajo. Podemos encontrar más información sobre estas observaciones y estrellas en diversas publicaciones del grupo de estrellas frías del Dpto de Astrofísica de la UCM (López-Santiago et al. 2010; Martínez-Arnáiz et al. 2010; Maldonado et al. 2010; Montes et al 2012)

### 1.3 Programa IRAF.

Todo el proceso de reducción de datos, incluido, evidentemente, el de normalización al continuo se realiza a través de las tareas incluidas en el paquete ECHELLE dentro del programa IRAF.

IRAF es una herramienta destinada al análisis y reducción de imágenes astronómicas. Es desarrollado y distribuido por la *National Optical Astronomy Observatories* (NOAO) que depende de la *Association of Universities for Research in Astronomy* (AURA) en colaboración con la *National Science Foundation*.

### 1.4 Base de datos Saada.

Por último vamos a crear una base de datos accesible desde internet en la cual estarán disponibles todos los espectros procesados en este trabajo con información propia de cada una de las estrellas como por ejemplo la metalicidad, la gravedad o la velocidad relativa respecto al Sol. En el tercer apartado de este trabajo explicaré de forma detallada cómo podemos desarrollar esta base de datos.

## 2. TRATAMIENTO DE ESPECTROS.

En esta segunda parte del trabajo explicaré de forma precisa cómo realizar el ajuste al continuo de todos los órdenes de un espectro ya reducido.

### 2.1 Supresión de rayos cósmicos.

Antes de iniciar el proceso de normalización al continuo hemos eliminado los rayos cósmicos que aparecían en los órdenes que debían ser normalizados.

Los rayos cósmicos son partículas altamente energéticas (con velocidades próximas a la de la luz) que viajan por nuestra Galaxia y alcanzan la superficie de la Tierra. Están formadas en un 90% por protones, en un 9% por partículas alfa y el resto son núcleos más pesados que el hidrógeno.

El origen de estos rayos cósmicos no está del todo claro. Se sabe que el Sol produce rayos cósmicos durante sus erupciones y a través del viento solar, pero ni su energía ni su cantidad son suficientes para explicar los rayos cósmicos observados desde tierra.

Cuando un rayo cósmico alcanza nuestro CCD observamos claramente como el píxel o los píxeles afectados producen una señal anormalmente alta. La aparición de estos rayos cósmicos en los CCDs es totalmente aleatoria pudiendo caer en las zonas ocupadas por el espectro de los objetos a estudiar destruyendo la información correspondiente a esa longitud de onda. Los rayos cósmicos son fácilmente detectables puesto que aparecen como un pico en el valor del flujo que destaca de forma clara tal y como vemos en la siguiente imagen.

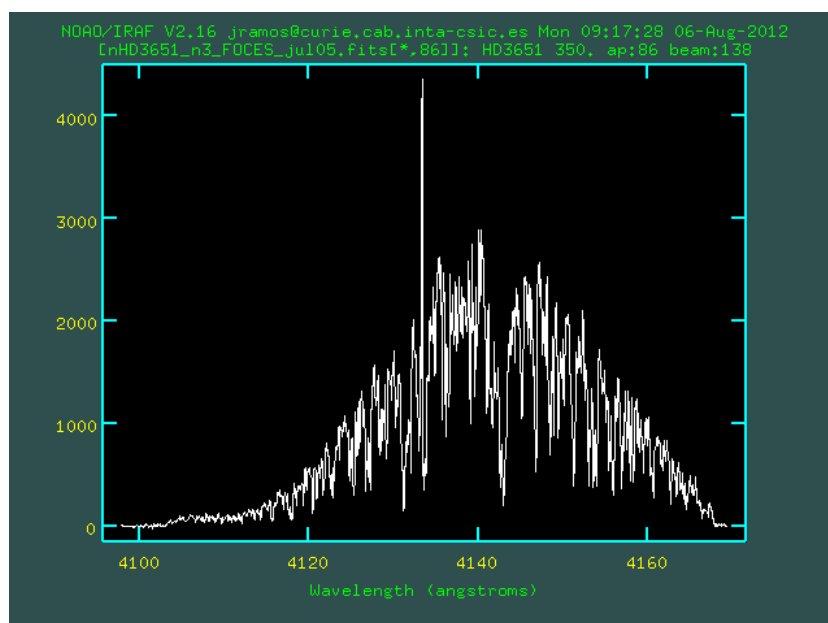


Fig. 1. Rayo cósmico.

Existen métodos para eliminar los rayos cósmicos utilizando técnicas de exposición como el *dithering* que consiste en tomar varias imágenes del mismo objeto desplazadas unas de otras mínimamente, para después hacer una estadística del flujo de cada píxel individual pudiendo desechar los píxeles que presenten un valor muy alejado de la media o la mediana. Para esto IRAF posee herramientas como *imcombine* que nos permite combinar imágenes para eliminar los rayos cósmicos o los píxeles calientes. Este proceso se realiza antes de la reducción del espectro.

Sin embargo, en nuestro caso los espectros fueron reducidos sin eliminar los rayos cósmicos por lo que tuvimos que eliminarlos manualmente estudiando todos los órdenes de cada espectro de forma detallada.

IRAF nos permite unir con una recta dos puntos distintos del espectro eliminando todo lo que hay entre medias. Para ello no tenemos más que pulsar la tecla “x” a cada lado de la zona que deseamos eliminar y después pulsar la tecla “r” para llevar a cabo la acción. El proceso paso a paso se ve claramente con la siguiente secuencia de imágenes.

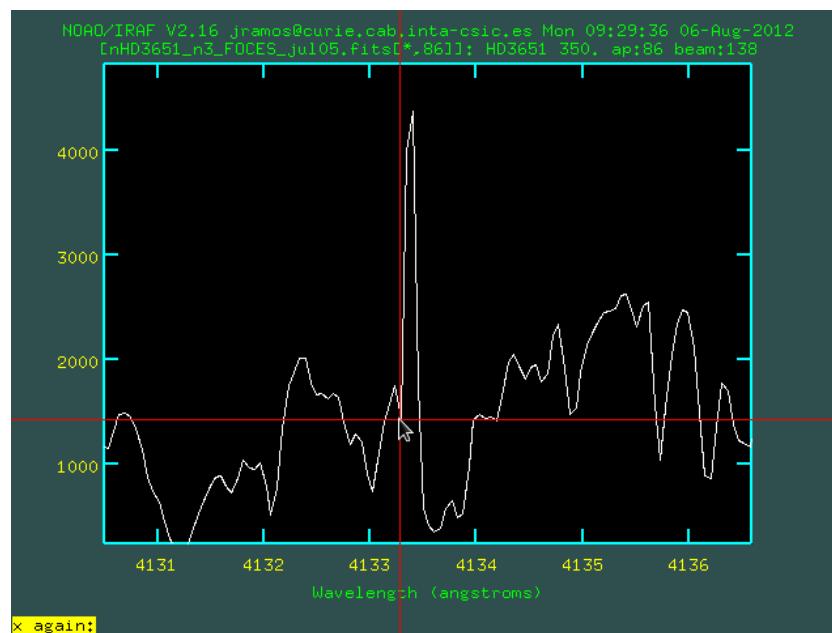


Fig. 2. Hacemos zoom en la imagen y pulsamos la “x” a uno de los lados del rayo cósmico.



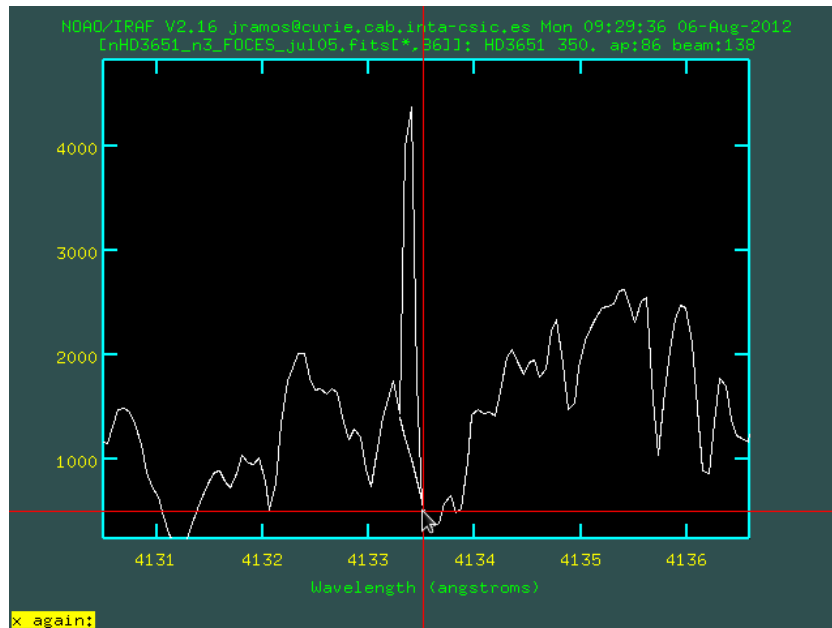


Fig. 3. Pulsamos la “x” al otro lado del rayo cósmico.

Finalmente pulsamos la tecla “r” y eliminamos el rayo cósmico tal y como se ve en la siguiente figura.

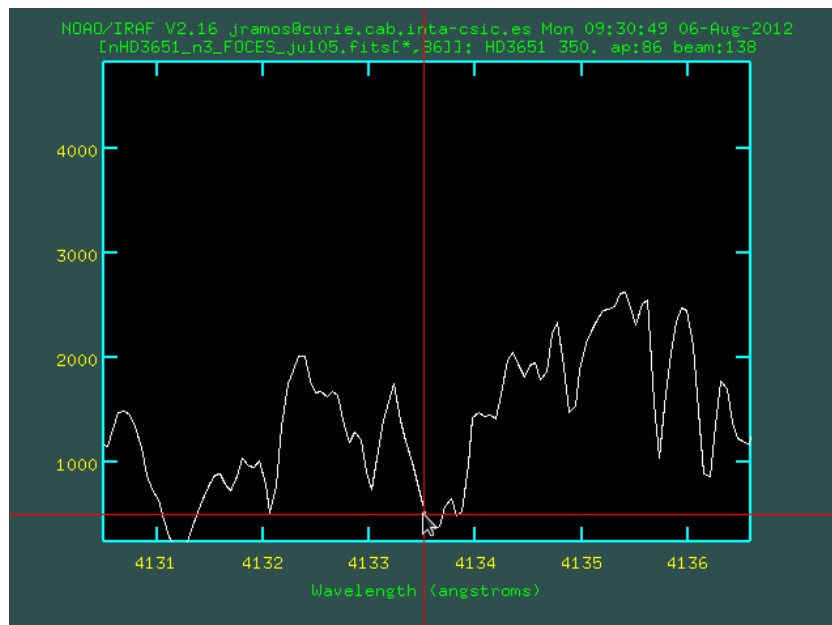


Fig. 4. Rayo cósmico corregido.

Una vez que quitamos los rayos cósmicos de cada orden debemos guardar el nuevo espectro corregido. Esto lo hacemos pulsando la letra “i” e introduciendo el nuevo nombre que queramos darle. En este caso será *NombreDelFichero\_L.fits*, añadimos la “L” al final para indicar que el espectro está “limpio” y no presenta rayos cósmicos.

## 2.2 Normalización al continuo.

El segundo paso que hay que llevar a cabo es la normalización al continuo de todos los órdenes del espectro. Para ello utilizamos la tarea “continuum” presente en el programa IRAF.

Esta tarea posee una gran cantidad de parámetros que podemos configurar para hacer nuestra normalización lo más precisa posible. La relación completa de los parámetros de la tarea “continuum” se recoge en el apéndice final. Ahora haremos referencia únicamente a los de mayor importancia.

Para configurar los parámetros de cualquier tarea en IRAF debemos teclear en la terminal “*epar NombreDeLaTarea*” en este caso “*epar continuum*”.

Lo primero que nos pide la tarea es que introduzcamos el nombre de la imagen que vamos a ajustar (*NombreDelFichero\_L.fits*) y el nombre de la imagen de salida, una vez ajustada, (*NombreDelFichero\_N.fits*). La “N” nos indica que esa imagen ya ha sido normalizada.

Después debemos indicar los órdenes que queremos normalizar, las imágenes originales tienen generalmente 111 órdenes, pero nosotros sólo nos quedaremos con aquellos que van del 10 al 94. Desechamos los primeros órdenes y los últimos debido a que presentan una gran cantidad de ruido y no son útiles para hacer ciencia.

El tipo de función con la que realizamos el ajuste viene por defecto y es de tipo “spline3”, podemos cambiar el orden de dicha función para mejorar el ajuste así como seleccionar los puntos que queremos que queden por debajo (*low\_rej*) o por encima (*high\_rej*) de la línea del continuo.

Aunque estos valores se pueden fijar de antemano lo mejor es adaptarlos a cada orden. Por eso es importante activar la opción que nos permite interaccionar con la tarea durante el ajuste (*interac = yes*).

Finalmente otro parámetro importante es *override* que nos impide volver a ajustar órdenes ya normalizados, ponemos, por tanto, *override = no*.

Una vez fijados todos los parámetros de la tarea *continuum* a nuestro gusto, tecleamos: go e iniciamos el ajuste del primer orden.

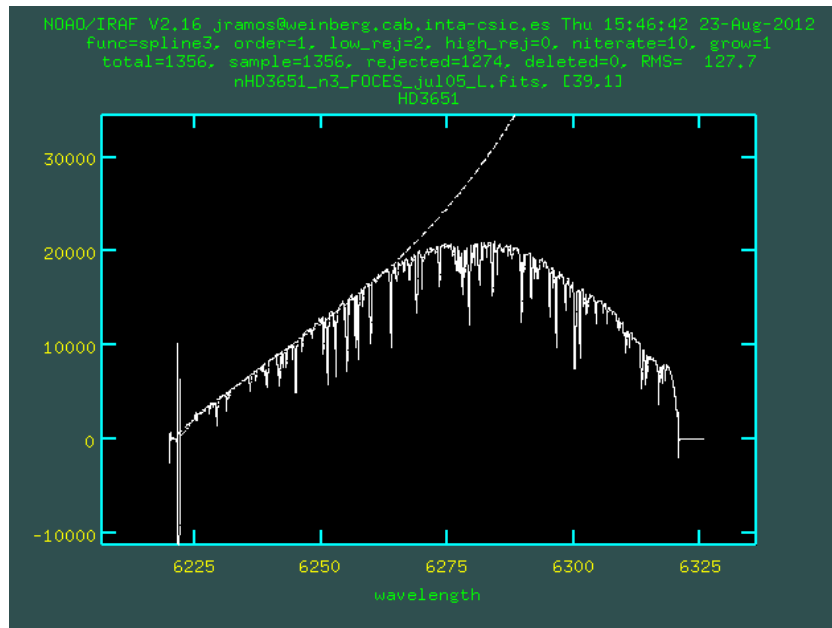


Fig. 5. Como vemos los extremos no deben ser tenidos en cuenta al realizar el ajuste.

Como vemos en la imagen hay zonas en los bordes de los órdenes que no deben ser tenidas en cuenta a la hora de realizar el ajuste, para ello utilizamos la tecla “s” que nos permite seleccionar la zona que queremos normalizar. También podemos utilizar esta herramienta para evitar que el programa ajuste regiones afectadas por rayos cósmicos que no hayan sido eliminados en el primer paso.

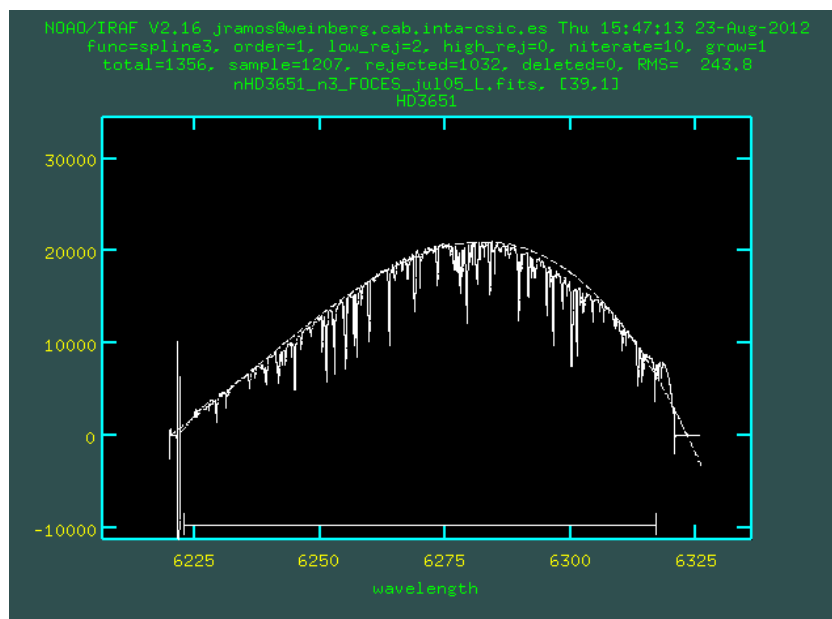


Fig. 6. Una vez seleccionada la zona vemos que el ajuste mejora claramente.

Como ya dijimos podemos mejorar el ajuste variando el orden de la función, esto se hace de forma interactiva tecleando: o y el número del orden. El orden más utilizado es el 3 aunque en ocasiones podemos utilizar órdenes superiores. Si bien, utilizar órdenes demasiado altos no es aconsejable puesto que la forma del continuo obtenida puede ser bastante irreal.

Otro elemento que hay que tener en cuenta en los órdenes del espectro es la presencia de ruido. Este ruido es más común en las longitudes de onda más cortas, para no ajustar los puntos producidos por el ruido utilizamos la herramienta: low que nos permite indicar la cantidad de puntos en unidades de sigma que dejamos por debajo del ajuste.

Una vez que hemos ajustado todos los parámetros y vemos que la función representa el continuo de forma adecuada podemos pasar al siguiente orden, para ello pulsamos la tecla q y ante la pregunta de si queremos pasar al siguiente orden tecleamos yes.

Cuando todos los órdenes están normalizados podemos calcular algunos parámetros de la estrella a partir de la anchura de las líneas y su profundidad equivalente.

### 2.3 Corte de los órdenes de los espectros.

Como ya hemos visto los órdenes presentan zonas en los bordes que debemos desechar, para ello utilizamos la tarea de IRAF “scopy”, esta tarea nos permite seleccionar los cortes dentro de un determinado orden para quedarnos sólo con la región debidamente normalizada.

Los principales parámetros dentro de “scopy” son: el nombre de la imagen de entrada, el nombre de la imagen de salida, los cortes en longitud de onda en unidades de angstroms ( $w1$  y  $w2$ ) y el orden al que debemos aplicar ese corte. El resto de parámetros se dejan como están por defecto.

Así, para cortar todos los órdenes de un espectro debemos aplicar un “scopy” a cada uno de ellos, cambiando en cada caso los cortes en longitud de onda. Este proceso es bastante tedioso, por eso hemos generado un “script” que nos permite hacer este trabajo de forma más rápida y cómoda.

Un “script” es una herramienta que nos permite realizar los cortes de todos los órdenes a la vez. Dentro de este “script” escribimos un “scopy” para cada orden con los cortes adecuados. Aunque los espectros tomados en una misma campaña de observaciones suelen tener los mismos cortes en longitud de onda, es importante revisar los valores de  $w1$  y  $w2$  durante el proceso de ajuste al continuo.

Por último debemos introducir el nombre de la imagen que deseamos cortar y su nombre de salida. Este “script” posee 84 “scopy”, por tanto, para evitar tener que cambiar los nombres de las imágenes de entrada y salida 84 veces cada vez que cambiemos de imagen, vamos a utilizar un “@file”.

Los “@file” son listas de objetos (imágenes en nuestro caso) que introducimos en el “scopy” como argumentos de entrada y de salida. Creamos así dos “@files”:

- @entrada: Que posee la imagen normalizada que queremos cortar.
- @salida: Que poseen el nombre de la imagen ya cortada (*NombreDelFichero\_c.fits*).

Una vez creado el “script” que en nuestro caso se llama “scopy\_varios\_N.cl” debemos ejecutarlo en la *xgterm* tecleando:

```
cl< scopy_varios_N.cl
```

Tras ejecutarlo obtenemos una imagen del espectro con todos los órdenes cortados.

A continuación mostramos un ejemplo de cómo queda uno de los órdenes del espectro que estamos tomando como ejemplo tras haber sido normalizado y cortado.

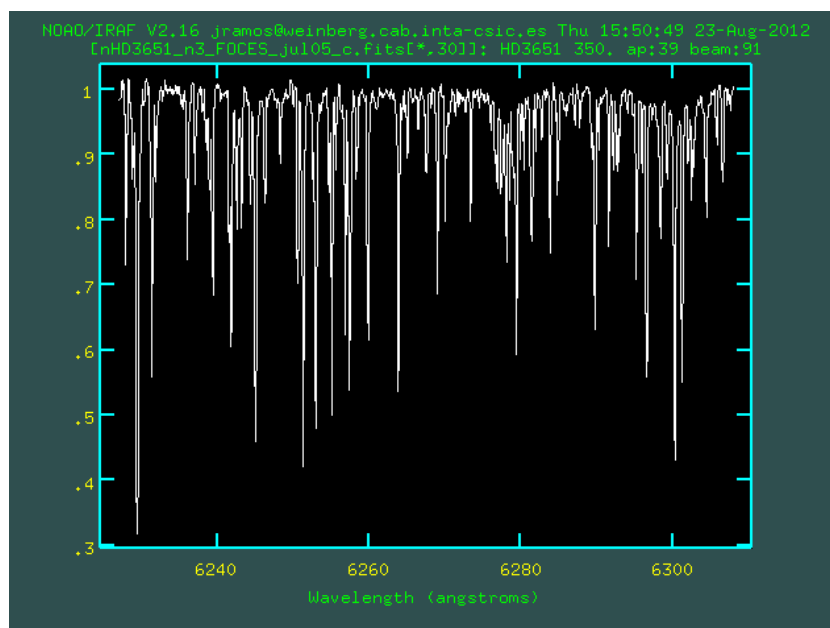


Fig. 7. Orden 39 una vez normalizado y cortado.

El “script” completo de “scopy\_varios\_N.cl” se encuentra en el apéndice final.

## 2.4 Pegado de los órdenes de los espectros.

En este paso vamos a utilizar la tarea “*scombine*” que nos permite pegar todos los órdenes del espectro para generar una imagen del espectro completo.

La tarea “*scombine*” es tremendamente compleja y resulta de gran utilidad cuando se necesitan combinar varias imágenes de un mismo objeto que han sido obtenidas con técnicas de observación como el *dithering* para posteriormente poder eliminar los rayos cósmicos, los píxeles calientes y demás defectos cosiméticos así como mejorar la señal ruido de las imágenes resultantes.

En nuestro caso el objetivo es simplemente unir todos los órdenes del espectro. Por ello en nuestro “script” eliminamos aquellas funciones asociadas a la eliminación de píxeles calientes y rayos cósmicos y le indicamos que en las zonas donde exista solapamiento haga una media de los valores que aparezcan en las distintas imágenes.

Por otro lado, a longitudes de onda largas, los órdenes no solapan de modo que en este caso aparece una línea horizontal uniendo los órdenes de los distintos espectros. Esto se observa en la siguiente imagen.

Como argumentos de entrada y salida también hemos creado dos “@files”:

- @salida\_n: Posee la imagen de entrada ya cortada y normalizada (*NombreDelFichero\_c.fits*).
- @salida\_pegado: Posee la imagen de salida ya pegada (*NombreDelFichero\_pegado.fits*).

Como en los casos anteriores el “script” completo se encuentra en el apéndice final con todos los detalles debidamente explicados.

El espectro pegado tiene la siguiente forma (ver también Fig. 11):

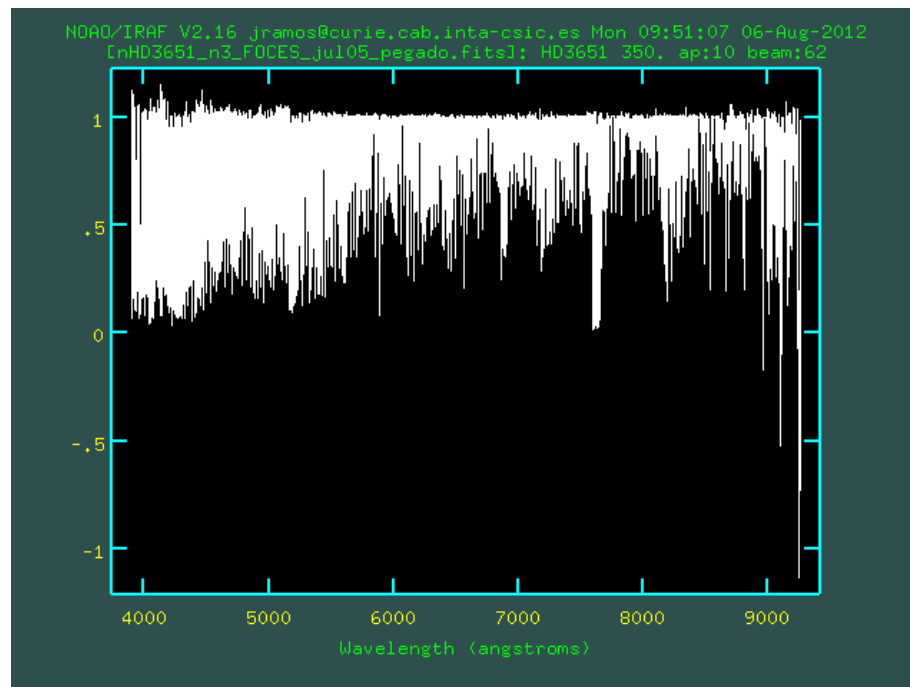


Fig. 8. Espectro con todos los órdenes pegados.

## 2.5 Corrección de la velocidad radial.

Por último debemos corregir el espectro del efecto Doppler.

Como ya sabemos el efecto Doppler consiste en un desplazamiento del espectro original de la estrella hacia longitudes de onda más cortas (azul) o longitudes de onda más largas (rojo) según la estrella se esté acercando o alejando de nosotros respectivamente con una determinada velocidad radial.

Para determinar esta velocidad utilizamos la tarea de IRAF "*fxcor*". Esta tarea realiza una correlación cruzada entre el espectro de la estrella problema con el espectro del Sol que tomamos como referencia ya que podemos considerar que está en reposo respecto a nosotros. Como parámetro de entrada también debemos indicar las regiones del espectro que queremos comparar. Como salida obtenemos un archivo ".txt" dentro del cual tenemos el valor de la velocidad radial a la que se mueve la estrella respecto a nuestros sistemas de referencias centrado en el Sol. Obtenemos sólo el valor de la velocidad radial porque es la componente de la velocidad responsable del efecto Doppler.

Los nombres de los ficheros que contienen el valor de la velocidad presentan la siguiente forma: "*NombreDelFichero\_velocidad.txt*".

Una vez obtenida la velocidad radial utilizamos la tarea “*dopcor*” para corregir de efecto Doppler el espectro de nuestra estrella. Es decir corregimos el desplazamiento del espectro en longitud de onda teniendo en cuenta  $V_r$

Esta tarea es bastante sencilla y tan sólo debemos introducir como parámetros de entrada la imagen “pegada” y la velocidad radial. La imagen de salida será:

“*NombreDelFichero\_final*”

Podemos confirmar si la corrección se ha realizado de forma adecuada viendo el valor de la longitud de onda en la que aparecen líneas características de absorción como, por ejemplo,  $H\alpha$  (6562.8 Å).

En el siguiente par de imágenes vemos el valor de la línea  $H\alpha$  para la imagen pegada sin corregir de efecto Doppler y para la imagen final ya corregida.

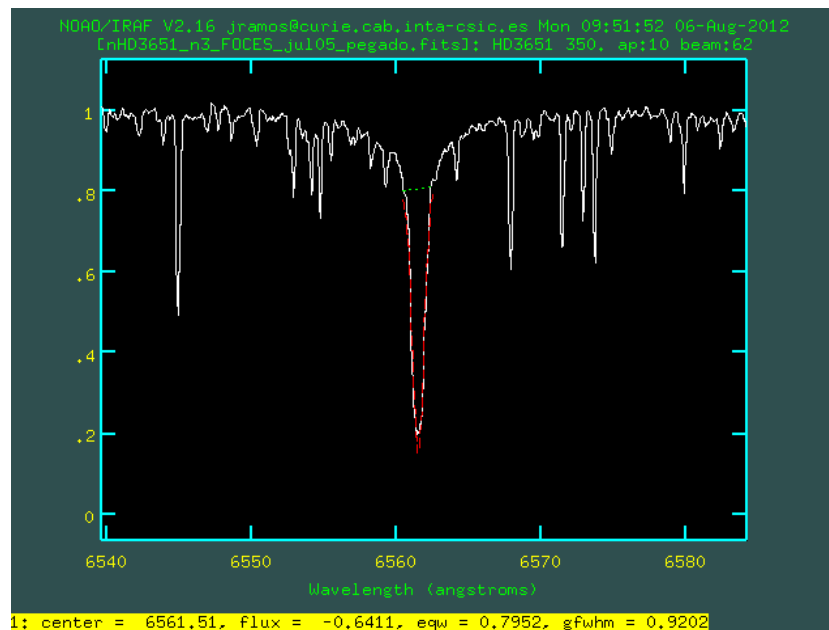


Fig. 9. Línea  $H\alpha$  centrada en 6561.51 Å, sin corregir.



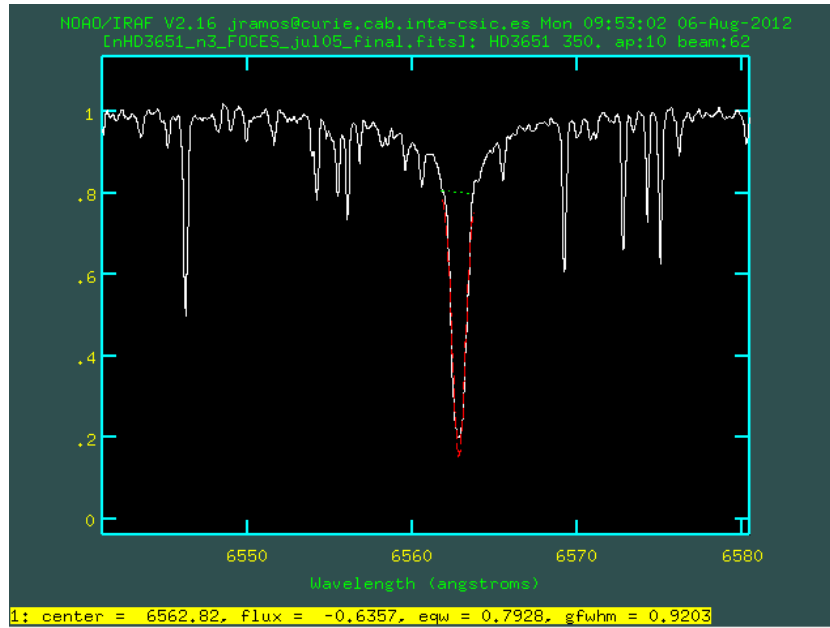


Fig. 10. Línea H $\alpha$  centrada en 6562.82 Å, ya corregida.

Una vez que tenemos los espectros procesados ya podemos utilizarlos para obtener algunos parámetros de las estrellas como son la metalicidad, la gravedad o la temperatura efectiva (utilizando el código *StePar*, ver Tabernero et al. (2012)).

En este sentido el grupo del Dr. David Montes utilizó los espectros finales de las estrellas HD 142267, HD 82885 y HD 3651 para obtener datos asociados a éstas y posteriormente presentarlos en la X Reunión Científica de la Sociedad Española de Astronomía (SEA) celebrada en Valencia (Montes et al. 2012) .

A continuación mostramos los espectros presentados junto con algunos de los parámetros calculados:

Nombre	STp	$T_{\text{eff}}$ (K)	$\log g$	$\xi$ (km/s)	[Fe/H]
HD 142267	G2V	$5768 \pm 35$	$4.42 \pm 0.08$	$1.00 \pm 0.05$	$-0.38 \pm 0.03$
HD 82885	G8V	$5536 \pm 37$	$4.43 \pm 0.09$	$1.32 \pm 0.06$	$0.27 \pm 0.03$
HD 3651	K0V	$5282 \pm 45$	$4.35 \pm 0.10$	$1.16 \pm 0.08$	$0.11 \pm 0.03$

El espectro final resultante de todo el proceso anteriormente descrito para el caso de estas tres estrellas se muestra a continuación:

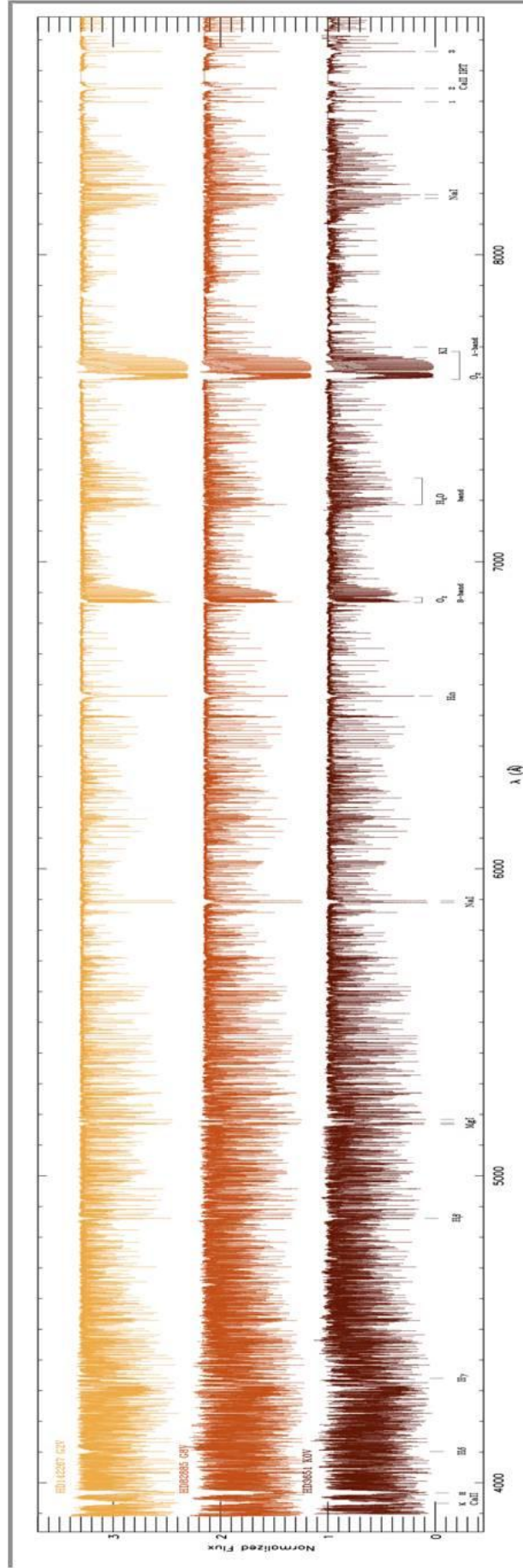


Fig. 11. Espectros finales para HD 142267, HD 82885 y HD 3651

En las Figs. 12-15 se pueden ver con más detalle algunas líneas importantes de estos espectros como: Ca II H y K, el triplete de Mg I b, el doblete de Na I D<sub>1</sub>, D<sub>2</sub> y H $\alpha$ .

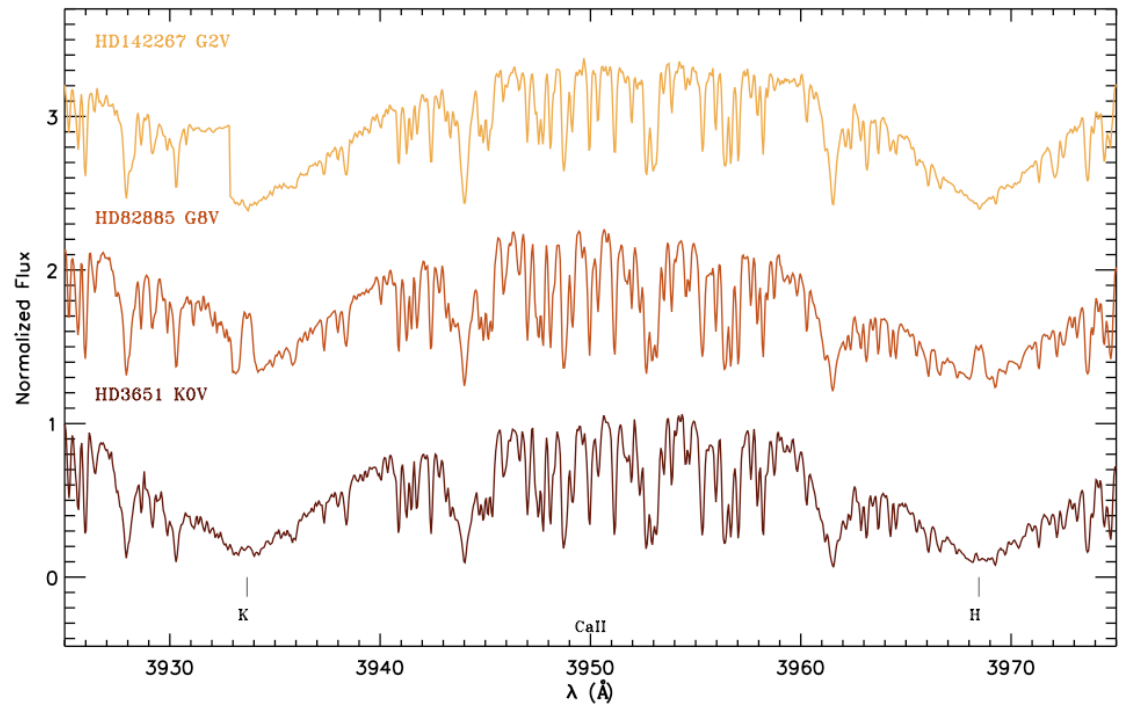


Fig. 12. Doblete de Ca II H y K.

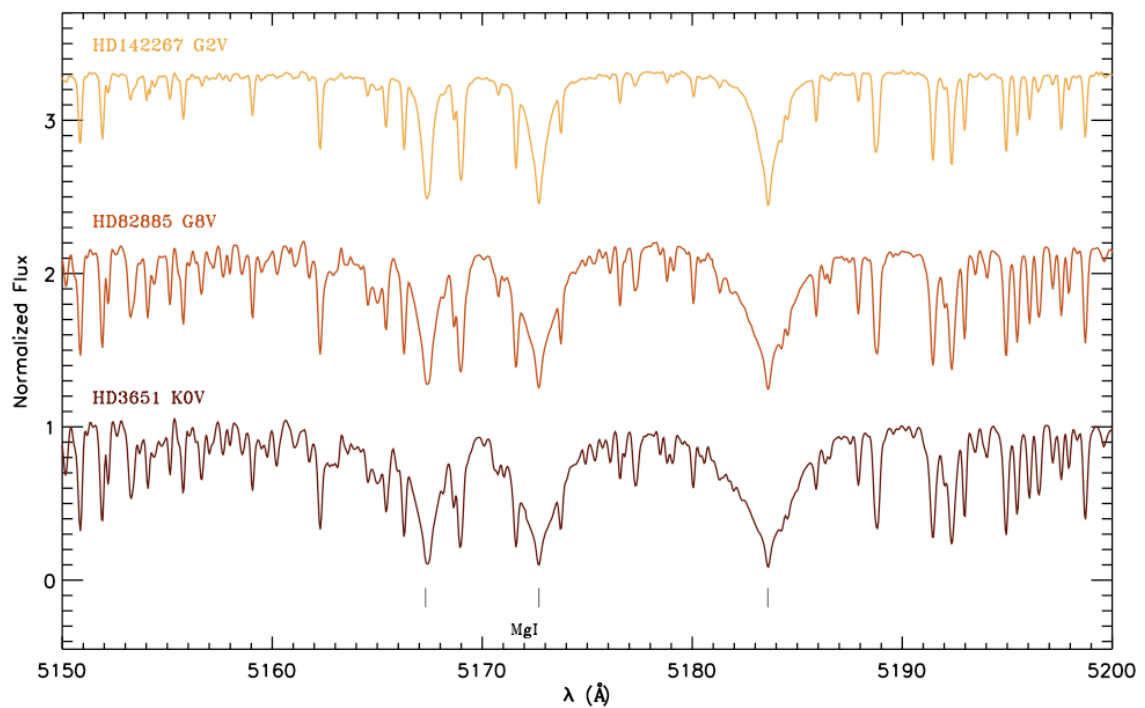


Fig. 13. Triplete de Mg I b1, 2, 3

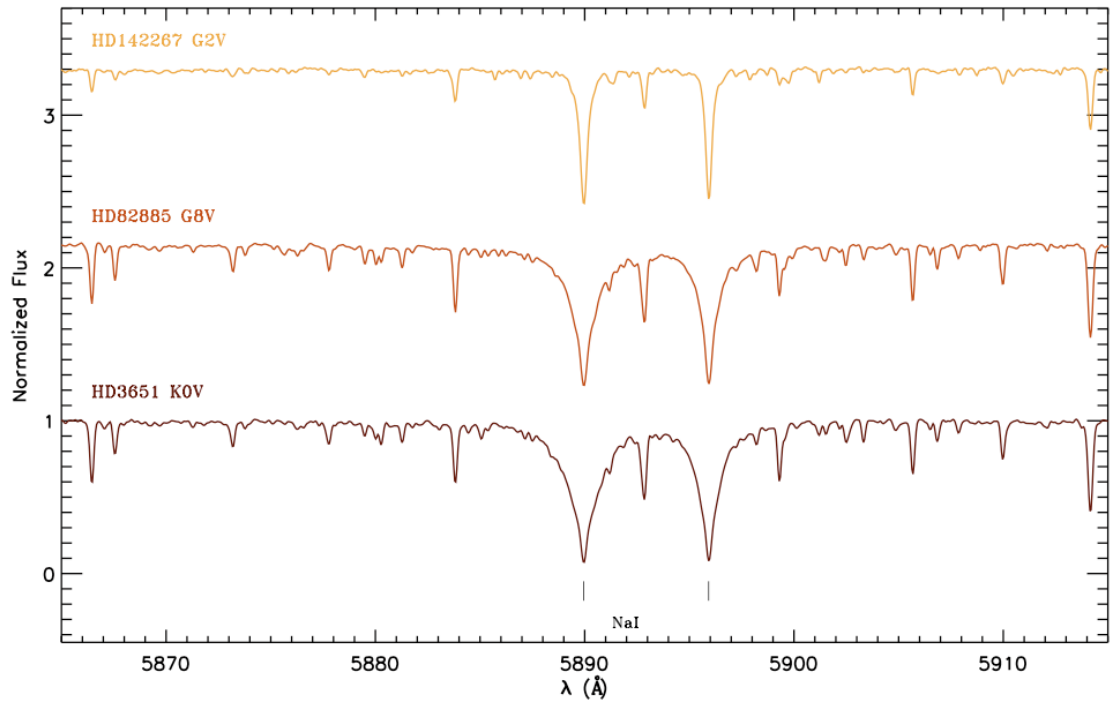


Fig. 14. Doblete de Na I D<sub>1</sub>, D<sub>2</sub>

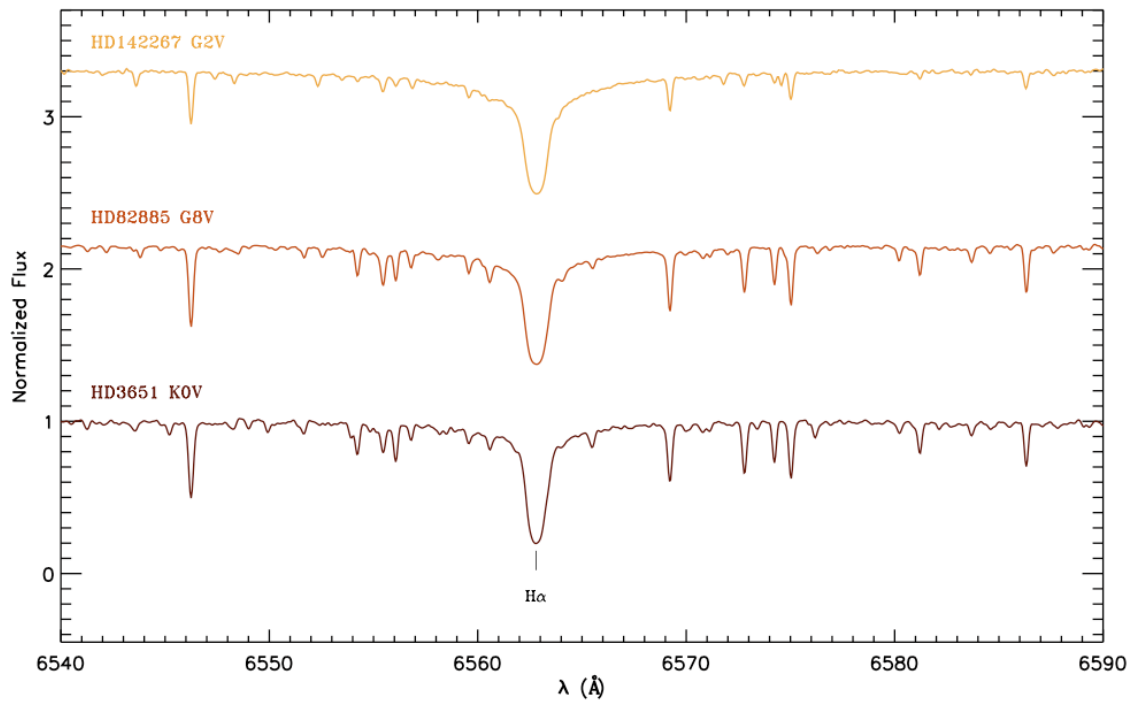


Fig. 15. Línea de H $\alpha$

### 3. BASE DE DATOS.

Tal y como dijimos en la introducción, la segunda parte de este trabajo consiste en generar una base de datos que sea accesible desde internet y permita a cualquier usuario trabajar con los espectros de las estrellas ya procesados.

Para generar esta base de datos utilizaremos el programa *SaadaDB* (<http://saada.u-strasbg.fr/saada/spip.php?article32>). Los pasos que debemos seguir para crear esta base de datos vienen detallados en la página web de Saada y los mostramos a continuación.

En primer lugar debemos crear en nuestro equipo un directorio que llamaremos “nuestro\_saada”, a su vez dentro de este directorio creamos otros cuatro llamados:

- Saada: Donde instalaremos el programa de Saada.
- Saadadb: Que servirá para alojar la base de datos.
- Repository: Directorio en el que meteremos los espectros que formarán la base de datos.
- Apache: Carpeta en la que instalaremos el programa Tomcat.

Una vez creados todos los directorios descargamos la versión de *Saada* ([Saada1.6.0](#)) desde su página web y la guardamos en la carpeta *saada*. Dentro de la terminal nos movemos hasta ese directorio y una vez allí ejecutamos:

```
java -jar Saada1.6.0.noplatform.jar
```

De esta forma se inicia el asistente de instalación de *Saada*. Elegimos de nuevo como carpeta de destino “*saada*” y aceptamos las condiciones de la licencia del producto para completar la instalación.

Una vez instalado el *Saada*, vamos a instalar el *Tomcat*. *Tomcat* es un servidor web que utilizará *Saada* para que se pueda acceder a la base de datos a través de internet.

La versión de *Tomcat* que debemos descargar para conseguir el mejor funcionamiento de *Saada* es la versión *Tomcat 6.0.x* que descargamos de la página oficial de *Tomcat*.

El programa que descargamos lo guardamos en la carpeta “*apache*” creada anteriormente y desde ese directorio nos movemos hasta la carpeta *bin* que está dentro del *Tomcat* una vez allí ejecutamos el siguiente comando:

```
./startup.sh
```

De esta forma ya hemos instalado y ejecutado *Tomcat*. Para comprobar que el proceso se ha realizado correctamente podemos introducir en nuestro navegador de internet la dirección <http://localhost:8080> y comprobar si aparece la siguiente página.

Una vez instalado el *Saada* y el *Tomcat* vamos a crear la base de datos propiamente dicha.

Para ello nos vamos a la carpeta *saada* y dentro del directorio *bin* tecleamos:

```
./newsaadadb
```

Se abre así una aplicación que nos va guiando en la creación de la base de datos. En primer lugar debemos darle un nombre, en nuestro caso será: *"esp\_coolstars"*.

También debemos indicar la carpeta en la que vamos a alojar la base de datos *"saadadb"* y la carpeta en la que se encuentran los espectros procesados *"repository"*.

En el siguiente paso debemos seleccionar el sistema con el que vamos a manejar la base de datos. En la página web de *Saada* se recomienda utilizar *SQLite* sin embargo esta opción nos dio algunos problemas técnicos, por lo que decidimos utilizar *MySQL* para poder trabajar con *MySQL* debemos crear primero una cuenta de administrador y otra de usuario.

Una vez hecho esto, el proceso de creación de la base de datos es igual al indicado en la página web. Uno de los últimos pasos es generar la URL que nos permitirá acceder a la base de datos desde internet.

El formato de la URL debe ser el siguiente:

```
http://HOSTNAME:TOMCATPORT/DATABASENAME
```

En nuestro caso es: [http://teide.fis.ucm.es:8080/esp\\_coolstars/](http://teide.fis.ucm.es:8080/esp_coolstars/)

Finalmente le indicamos que instale la base de datos.

El siguiente paso es cargar los espectros a la base de datos. Para ello nos movemos hasta el directorio *bin* dentro de la carpeta *saada* y ejecutamos:

```
./saadmintool
```

De esta forma se nos abre una ventana de administración desde la cual cargar los espectros.

Creamos así una nueva colección a la que llamamos *Spectra* . Si pinchamos sobre el icono de esta colección vemos que existen varias “subcarpetas” una de ellas se llama *TABLE*.

Pinchando con el botón derecho sobre esta subcarpeta podemos seleccionar la opción “*Load Data*” y seleccionar los archivos “*.fits*” que formarán parte de la colección. Una vez seleccionados los cargamos pinchando en el botón “*Load Files*” para finalmente confirmar la carga pinchando en “*Run Data Loader*” .

De esta forma conseguimos cargar los espectros a la base de datos que después están disponibles a través de internet. Desde la página web se podrán descargar también los parámetros estelares calculados con los modelos.

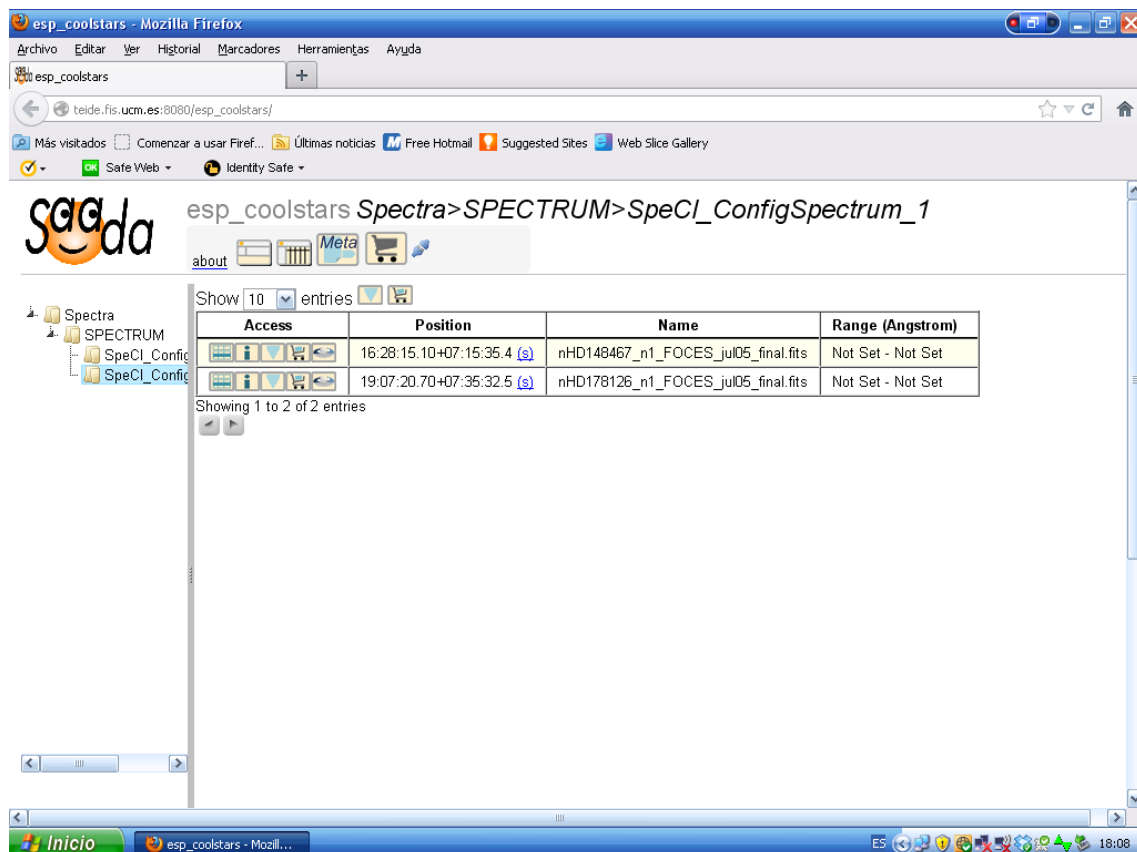


Fig. 16 Interfaz de la base de datos.

#### 4. BIBLIOGRAFÍA

- López-Santiago J., “Estudio de la actividad, rotación, cinemática y edad en estrellas frías miembros de grupos cinemáticos jóvenes”. Tesis doctoral. Departamento de astrofísica. Universidad Complutense de Madrid. 2005.
- López-Santiago J., Montes D., Crespo-Chacón I., Fernández-Figueroa M.J. “The nearest young moving groups” 2006, ApJ, 643, 1160.
- López-Santiago J., Montes D., Gálvez-Ortiz M.C. et al. “A high-resolution spectroscopic survey of late-type stars: chromospheric activity, rotation, kinematics, and age” 2010, A&A, 514, A97.
- Maldonado J., Martínez-Arnáiz R.M., Eiroa C., Montes D., Montesinos B. “A spectroscopy study of nearby late-type stars, possible members of Stellar Kinematic Groups” 2010, A&A, 521, A12.
- Martínez-Arnáiz R.M., Maldonado J., Montes D. et al. “Chromospheric activity and rotation of FGK stars in the solar vicinity. An estimation of the radial velocity jitter” 2010, A&A 520, A79.
- Martínez Arnáiz R.M. “Chromospheric activity and rotation of FGK stars in the solar neighbourhood: characterizing possible exoplanetary system host stars”. Tesis doctoral. . Departamento de astrofísica. Universidad Complutense de Madrid. 2011.
- Montes D, Tabernero H.M, González Hernández J.I, “Chemical tagging of FGK stars: testing membership to stellar kinematic groups”. Proceedings of the The metallicity distribution in the Milky Way discs, GREAT-ESF Workshop. 2012.
- Montes D, et al. “Cool stars: spectral library of high-resolution echelle spectra and database of stellar parameters”. Proceedings of the X Reunión Científica de la SEA, Sociedad Española de Astronomía (9 - 13 Julio 2012, Valencia, Spain).
- Tabernero H.M, Montes D., González Hernández J.I, “Chemically tagging the Hyades Supercluster: A homogeneous sample of F6-K4 kinematically-selected ”, Astronomy & Astrophysics, A&A, in press <http://arxiv.org/abs/1205.4879>. 2012.



Enlaces web.

- <http://iraf.noao.edu/>
- <http://saada.u-strasbg.fr/saada/spip.php?article32>

## 5. APÉNDICE

En este apéndice final vamos a incluir el código completo de todos los “scripts” y tareas utilizadas durante el trabajo

### 5.1 Lista de estrellas.

A continuación mostramos una lista completa con todos los nombres de las estrellas, así como el tipo espectral, el instrumento utilizado y el año en que se realizaron las observaciones:

HIP	HD	Tipo espectral	Espectrógrafo	Campaña de observación
78072	142860	F6IV	FOCES	2005
17147	22879	F9V	FOCES	2006
77801	142267	G0V	FOCES	2009
1499	1461	G0V	FOCES	2005
48113	84737	G0,5V	FOCES	2002
7918	10307	G1.5V	FOCES	2001
95319	182488	G8V	FOCES	2005
47080	82885	G8V	FOCES	2006
84195	155712	K0	FOCES	2005
112190	215152	K0	FOCES	2005
3093	3651	K0.5V	FOCES	2005
7981	10476	K1V	FOCES	2005
3765	4628	K2V	FOCES	2006
105152	202751	K2V	FOCES	2005
114886	219538	K2V	FOCES	2005
114622	219134	K3/K4V	FOCES	1999
113718	217580	K4V	FOCES	2005
12929	17230	K5	FOCES	2006
83591	154663	K5V	FOCES	1999
93871	178126	K5V	FOCES	2005
104214	201091	K5V	FOCES	1999
80644	148467	K6V	FOCES	2005
104217	201092	K7V	FOCES	1999
60661	-	M0V	FOCES	2002
37826	62509	K0III	FOCES	2007
98337	189319	M0III	FOCES	2002
57939	103095	G8V	FOCES	2007

## 5.2 “scopy\_varios\_N”

Como ya comentamos este “script” está formado por un “scopy” por cada orden. En cada “scopy” cambiamos los cortes.

```
scopy ("@entrada",  
"@salida", w1=INDEF, w2=INDEF, apertures="10-94", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada",  
"@salida", w1=9150., w2=9270., apertures="10", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada",  
"@salida", w1=9000., w2=9125., apertures="11", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=8860., w2=8980., apertures="12", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=8728., w2=8843., apertures="13", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=8600., w2=8712., apertures="14", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,
```

merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=8470., w2=8564., apertures="15", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=8340., w2=8452., apertures="16", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=8221., w2=8326., apertures="17", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=8105., w2=8200., apertures="18", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7990., w2=8090., apertures="19", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7875., w2=7975., apertures="20", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7770., w2=7863., apertures="21", bands=" ", beams="",

apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7664., w2=7764., apertures="22", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7560., w2=7665., apertures="23", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7467., w2=7553., apertures="24", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7364., w2=7457., apertures="25", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7271., w2=7363., apertures="26", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=7176., w2=7271., apertures="27", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

```
scopy ("@entrada", "@salida", w1=7090, w2=7177., apertures="28", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=7004., w2=7090., apertures="29", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6917., w2=7007., apertures="30", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6838., w2=6923., apertures="31", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6750., w2=6836., apertures="32", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6676., w2=6752., apertures="33", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6596., w2=6674., apertures="34", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6520., w2=6600., apertures="35", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6446., w2=6524., apertures="36", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6373., w2=6451., apertures="37", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6300., w2=6382, apertures="38", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6235., w2=6310., apertures="39", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6167., w2=6240., apertures="40", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=6095., w2=6178., apertures="41", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,
```

merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=6036., w2=6102., apertures="42", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5972., w2=6041., apertures="43", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5915., w2=5980., apertures="44", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5848., w2=5918., apertures="45", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5790., w2=5860., apertures="46", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5727., w2=5800., apertures="47", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=5675., w2=5745., apertures="48", bands=" ", beams="",



```
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5617., w2=5684., apertures="49", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5562., w2=5628., apertures="50", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5510., w2=5576., apertures="51", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5455., w2=5520., apertures="52", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5405., w2=5472., apertures="53", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5350., w2=5420., apertures="54", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5300., w2=5370., apertures="55", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5252., w2=5317., apertures="56", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5208., w2=5271., apertures="57", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5161., w2=5217., apertures="58", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5115., w2=5174., apertures="59", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5075., w2=5128., apertures="60", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=5020., w2=5083., apertures="61", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4980., w2=5037, apertures="62", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4936., w2=4996., apertures="63", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4890., w2=4952., apertures="64", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4848., w2=4907., apertures="65", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4814., w2=4869., apertures="66", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4765., w2=4822., apertures="67", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4729., w2=4787., apertures="68", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,
```

merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4695., w2=4749., apertures="69", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4655., w2=4707., apertures="70", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4615., w2=4670., apertures="71", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4580., w2=4632., apertures="72", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4544., w2=4596., apertures="73", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4506., w2=4560., apertures="74", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4467., w2=4525., apertures="75", bands=" ", beams="",

apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4440., w2=4487., apertures="76", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4400., w2=4455., apertures="77", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4370., w2=4420., apertures="78", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4330., w2=4384., apertures="79", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4305., w2=4352., apertures="80", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

scopy ("@entrada", "@salida", w1=4280., w2=4320., apertures="81", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)

```
scopy ("@entrada", "@salida", w1=4240., w2=4287., apertures="82", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4205., w2=4245, apertures="83", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4180., w2=4220., apertures="84", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4155., w2=4190., apertures="85", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4120., w2=4160., apertures="86", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4093., w2=4132., apertures="87", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4070., w2=4105., apertures="88", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4040., w2=4070., apertures="89", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=4001., w2=4040., apertures="90", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=3981., w2=4018., apertures="91", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=3951., w2=3988., apertures="92", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=3924., w2=3959., apertures="93", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("@entrada", "@salida", w1=3895., w2=3924., apertures="94", bands=" ", beams="",  
apmodulus=0, format="multispec", renumber=no, offset=0, clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

### 5.3 “scombine\_varios\_N”

Como ya hemos comentado este “script” se limita a unir todos los órdenes. En las zonas de solapamiento, calcula una media del flujo y asigna ese valor al píxel. Esta opción se selecciona al poner: *combine="average"* .

```
scombine ("@salida_n","@salida_pegado", noutput="", logfile="STDOUT",
apertures="", group="all",combine="average", reject="none", first=no, w1=INDEF,
w2=INDEF, dw=INDEF,nw=INDEF,scale="none", zero="none", weight="none",
sample="",lthreshold=INDEF, hthreshold=INDEF, nlow=1, nhigh=1, nkeep=1,
mclip=yes,lsigma=3., hsigma=3., rdnoise="0.", gain="1.", snoise="0.",
sigscale=0.1,pclip=-0.5, grow=0, blank=1.)
```

### 5.4 *epar continuum*

Esta tarea fue utilizada para normalizar todos los órdenes de los espectros, a continuación mostramos todas las opciones de la tarea con una breve explicación de cada una:

PACKAGE = onedspec

TASK = continuum

input =	Input images
output =	Output images
(lines =	*) Image lines to be fit
(bands =	1) Image bands to be fit
(type =	ratio) Type of output
(replace=	no) Replace rejected points by fit?
(wavesca=	yes) Scale the X axis with wavelength?
(logscal=	no) Take the log (base 10) of both axes?
(overrid=	no) Override previously fit lines?
(listonl=	no) List fit but don't modify any images?
(logfile=	logfile) List of log files



(interac=	yes)	Set fitting parameters interactively?
(sample =	*	Sample points to use in fit
(naverag=	1)	Number of points in sample averaging
(functio=	spline3)	Fitting function
(order =	1)	Order of fitting function
(low_rej=	2.)	Low rejection in sigma of fit
(high_re=	0.)	High rejection in sigma of fit
(niterat=	10)	Number of rejection iterations
(grow =	1.)	Rejection growing radius
(markrej=	yes)	Mark rejected points?
(graphic=	stdgraph)	Graphics output device
(cursor =	)	Graphics cursor input
ask =	yes	
(mode =	ql)	

### 5.5 *epar dopcor*

Con esta tarea, una vez que sabemos la velocidad relativa entre la estrella observada y el Sol, podemos corregir el espectro de efecto Doppler.

PACKAGE = onedspec

TASK = dopcor

input =	List of input spectra
output =	List of output spectra
redshift=	Redshift or velocity (Km/s)
(isveloc=	no) Is the redshift parameter a velocity?
(add =	no) Add to previous dispersion correction?
(dispers=	yes) Apply dispersion correction?
(flux =	no) Apply flux correction?
(factor =	3.) Flux correction factor (power of 1+z)
(apertur=	) List of apertures to correct
(verbose=	no) Print corrections performed?
(mode =	ql)